Design of Low Cost Multi Channel Data Acquisition System for Meteorological Application

A thesis submitted in partial fulfillment of the requirements for the award of the degree for award of the degree

of

Master of Technology

Electronics and Communication Engineering

(Specialization: Electronics and Instrumentation Engineering)

by

Nisha Kashyap

Roll Number: 213EC3219

Under the Supervision of

Prof. Umesh Chandra Pati

Department of Electronics and Communication Engineering

National Institute of Technology, Rourkela, Odisha

May 2015

CERTIFICATE

This is to certify that the thesis entitled **DESIGN OF LOW COST MULTI CHANNEL DATA ACQUISITION SYSTEM FOR METEOROLOGICAL APPLICATION** submitted by **NISHA KASHYAP** bearing roll no. 213EC3219 in partial fulfilment of the requirements for the award of Master of Technology in Electronics and Communication Engineering with specialization in Electronics and Instrumentation Engineering during session 2013-2015 at National Institute of Technology, Rourkela, Odisha is a record of bona fide research work under my supervision.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University or Institute for the award of any Degree or Diploma.

Prof. Umesh Chandra Pati
Associate Professor
Department of Electronics and Communication Engineering
National Institute of Technology
Rourkela
Odisha

ACKNOWLEDGEMENTS

First of all, I would like to express my deep sense of respect and gratitude towards my supervisor Prof. Umesh Chandra Pati, who has been the guiding force behind this work. I am greatly indebted to him for his constant encouragement, invaluable advice and for propelling me further in every aspect of my academic life. He has been a wonderful mentor all throughout my stay in NIT. I consider it my good fortune to work with such a wonderful person who has had a profound effect on my career and my outlook towards the future.

Next, I want to express my respects to Prof. A.K. Swain, Prof. T. K. Dan, Prof. D.P. Acharya, Prof. S.K. Patra, Prof. S. Meher for teaching me and helping me how to learn. They have always been very helpful and resourceful in all difficult situations and for that I am grateful to them.

I would like to thank Lab Assistance Mr. Aditya and Mrs. Momita. My sincere thanks to my PhD sir Mr. Subhransu and Sankat for their enthusiastic encouragement, accurate feedbacks and invaluable suggestions. I would also like to thank and appreciate my friends and especially my classmates for all the enjoyment that we have had in the last two years along with serious studies and hard work. They have helped me become a better student and a better person. Their company has made my stay at NIT, Rourkela very memorable.

I am especially indebted to my parents for their love, sacrifice, and support. They have been my role models since childhood and have set great examples for me about how to live, study and work.

Nisha Kashyap

Rourkela

Abstract

The primary objective of the present work is to design a low cost multi channel data acquisition system which can be used for meteorological application. The proposed multi channel data acquisition system acquires ambient temperature, barometric pressure, altitude, humidity and light intensity data from environment and stores the data for future use. The sensors are connected with a low cost microcontroller (ATmega328) unit which performs the data acquisition and data logging operation. A real time clock is used to keep current time with the measurement. Different communication interfaces such as serial communication, wireless communication and ethernet communication are used in the data acquisition system. In serial communication, the sensor data are logged in COM port of PC. LabVIEW based application is developed which provides graphical user interface for the user. VISA protocol is used to communicate the COM port data with LabVIEW. Other communication protocol such as wireless communication and ethernet communication protocol is used to transmit the sensor data over a communication channel. In ethernet communication, the data is uploaded in ethernet which can be viewed using a web browser. In wireless communication Xbee transmitter and Xbee receiver modules are used to transmit data over a longer distance. The salient features of this developed system is that the system is low cost, uses open source softwares like arduino and python and the system is economical (i.e INR 6500). This developed system performs satisfactorily under different condition. The system is tested for 24 hour in April 2015 and provides satisfactory results.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Overview

Meteorology is an interdisciplinary science, which studies about environment, atmosphere. To study the current behaviour of atmosphere and predict the future behavior of atmosphere, different atmospheric parameters are measured and analyzed. Data acquisition and data logging of environmental variables are a real challenge and prediction of future behavior of environment from the present data is also a new challenge for the scientists. Some of the well-known meteorological parameters are ambient temperature, relative humidity, wind speed, barometric pressure, wind direction and atmospheric gas components.

1.2 Literature Review

1.2.1 PC based Data Acquisition

Greenhouse monitoring using microcontroller based meteorological data acquisition system is developed in [1]. During multi-channel data acquisition several sensors are connected to the microcontroller or PC. Sensor data should be stored in a memory unit. Now a days high speed real time multi-channel data acquisition system is available with transmission rate up to 3 GigaSamples/Sec. For this kind of high data rate Field Programmable Gate Array (FPGA) and Advanced RISC Machine (ARM) processor are used [2]. A multi-channel data acquisition system is designed using PIC 16F877 microcontroller. The PIC16F877 is interfaced with FLASH memory using serial peripheral interface (SPI) communication. This approach stores all the data in FLASH memory which is easily accessible [3].

1.2.2 Wireless Data Acquisition

Microcontroller based data acquisition system for weather station monitoring is discussed in [4]. In this data acquisition system, sensor data are transmitted to GSM modem every 24 hour via RS-232 interface. Low cost wireless autonomous remote weather monitoring system is developed in [5]. This system collects data and transfers the data to a PC and a remote server using wireless technique.

Data acquisition system in renewable energy sources is one of the prime importance. In [6], Zigbee based data acquisition system is developed for online monitoring of grid connected photovoltaic (PV) system . M. Funetes et al., designed a low cost autonomous data logger for PV system monitoring using microcontroller [7]. Wireless based measurement system to determine quality of water is developed in [8]. In [9], wireless sensor network is developed to monitor and control agricultural field. A review of wireless sensor network in food industry is provided in [10]. Fuzzy logic control is applied in irrigation system where the data comes from the wireless sensor network employed in agricultural field. The fuzzy logic control provides accurate control logic for irrigation system such as flow control, sprinkle control, temperature control [11]. In [12], GPRS based data acquisition and control is designed. In GPRS based acquisition, the data is transmitted through mobile phone. Wireless data acquisition system is also used in monitoring construction activity which is illustrated in [13]. In [14], the authors developed low cost wireless system for weather monitoring.

1.2.3 Web Based Data Acquisition System

Design of web based data acquisition system is developed in [15]. In the web based data acquisition system, the entire data acquisition, monitoring and control is done using HTTP server. SCADA system for oil refinery system is developed in [16]. The SCADA system is built upon a PLC and DCS platform and it is web based. Therefore, the SCADA system plays an important role in monitoring and control of critical processes like oil refinery.

1.2.4 Open Source Data Acquisition System

Open source multi functional DAQ system is developed in [17]. The open source DAQ uses open source hardware and software platforms like microcontroller and pyton for development of data acquisition system. The designed DAQ is compared with NI-USB 6008.

Data logging of a boiler plant is discussed in [18]. In this paper, temperature and water level of one of the boiler from three boilers is monitored continuously using a LabVIEW based application. The data is stored in a file, which can be used in future. In [19], STC89C52 is used to develop multi-channel data acquisition system. CD4067 is used as channel selector and a high precision A/D converter (AD574A) is used. Data acquisition system for accurate predication of Tsunami is developed by [20]. In this system, the sensors are placed in deep sea.

1.3 Motivation

With rapid advances in VLSI and MEMS techniques, the size of the sensors are getting smaller and functionality of the sensors are increasing. Signal conditioning techniques are inbuilt in the sensor itself and the sensors provide calibrated output which can be directly interfaced with a microcontroller. The cost of single unit of MEMS based sensors are also moderate to low which allows user to use these sensors in variety of fields with much ease. Different communication protocol such as inter integrated circuit (I2C), SPI has revolutionized the sensor design process. Like sensors, the design of microcontroller has also seen a sea change due to rapid development in VLSI technology. In older microcontrollers like ATMEL 89C51, additional ADC is required and the microcontroller don't have I2C protocol or SPI protocol neither PWM pins. Now a days ATmega328 microcontroller has in built ADC with 10 bit resolution and I2C protocol, SPI protocol available. With SPI protocol, different communication protocols such as GSM, ethernet and GPRS can be interfaced with the microcontroller. The microcontroller output can be interfaced with National Instrument's LabVIEW using VISA protocol.

The main motivation of this thesis is to design a low cost multi channel data acquisition system with different communication protocol. The multi channel data acquisition system is used in environmental monitoring.

1.4 Objective

- To design and implement multi channel data acquisition system using serial communication protocol
- To design and implement multi channel data acquisition system using wireless communication protocol
- To design and implement multi channel data acquisition system using ethernet communication protocol
- Test the designed system in real world environment using all the communication protocol
- The designed system should be of low cost and built using open source platform

1.5 Organization of Thesis

This thesis has 5 chapters. Chapter 1 provides basic introduction of the thesis. Chapter 2 provides details about data acquisition system. Chapter 3 presents the hardware design details of multi channel data acquisition system. Chapter 4 provides results and discussion and Chapter 5 concludes the thesis.

CHAPTER 2

Data Acquisition System

A generalized data acquisition system consists of different components such as sensing unit, signal conditioning unit, computing unit and display unit. The data acquisition system (DAQ) acquires particular environmental or physical parameters from the real world, performs signal conditioning on it and computes the actual value for display purpose. The block diagram of measurement system is illustrated in Fig. 2.1.

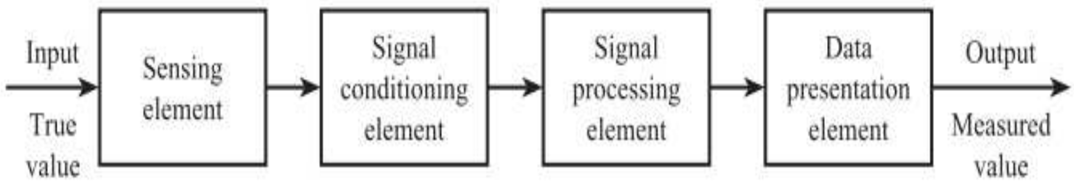

Figure 2.1: Block diagram of measurement system

2.1 Types of Data Acquisition System

Data acquisition system (DAQ) can be broadly classified in to three different categories such as

- PC based DAQ

- DAQ based on independent acquisition instruments
- Modular DAQ

Fig. 2.2 shows the block diagram of PC based DAQ, Fig. 2.3 presents the DAQ based on independent acquisition instrument and Fig. 2.4 shows the modular DAQ.

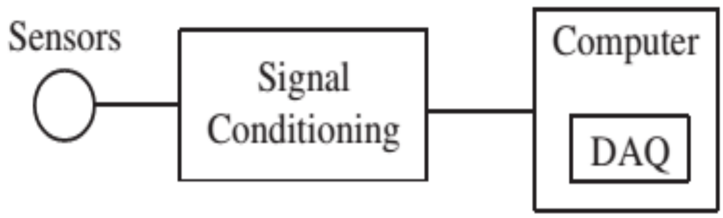

Figure 2.2: Block diagram of measurement system

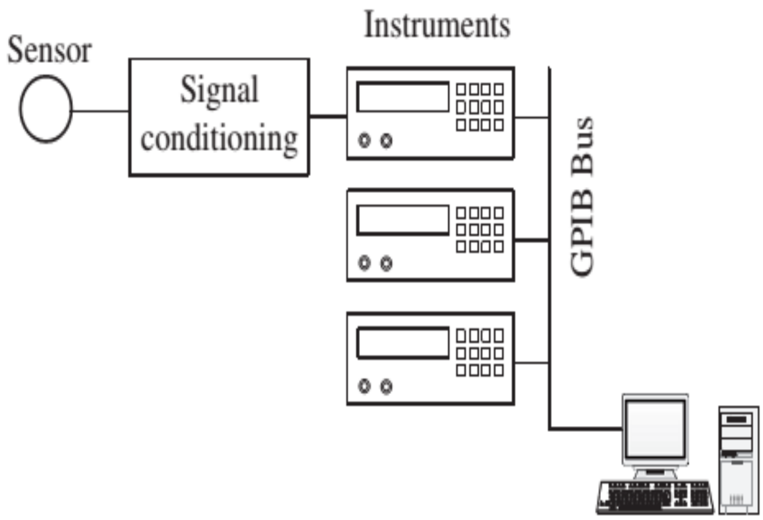

Figure 2.3: Block diagram of measurement system

2.2 Functions of Data Acquisition System

A DAQ system is capable of performing the following four important functions required for measurement and control application.

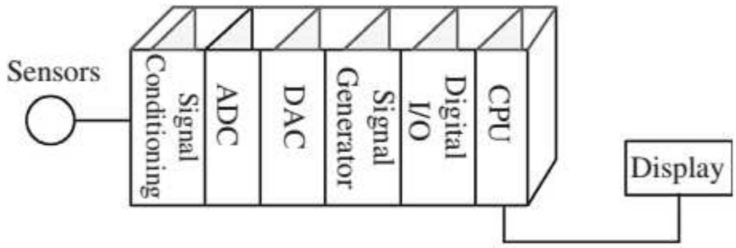

Figure 2.4: Block diagram of measurement system

2.2.1 Analog to digital conversion

For measurement application ,the DAQ system converts the analog signal from signal conditioning circuits to a digital data and transfers it to the computer.

2.2.2 Digital to analog conversion

For control application ,it converts digital data from the computer to analog signals for controlling analog devices.

2.2.3 Digital input/output

A DAQ system supporting digital input output function can receive and transfer digital data between PC and external digital device.

2.2.4 Timing I/O

Timing I/O function of a DAQ system is used for controlling the timing of data conversion and timing of data transfer between the pc and external device.

2.3 Data Acquisition Configuration

There are different data acquisition configurations commercially available. The list of different data acquisition configuration are shown below

- Local data acquisition system
  - Plug-in data acquisition system
  - Parallel port data acquisition system

- Data acquisition system using serial interface
  - UART
  - USB
  - IEEE-1394
  - I2C, SPI
- Networked data acquisition system
  - Analog transmission
  - Digital transmission
  - Hybrid transmission
- GPIB data acquisition system

CHAPTER 3

Hardware Design of Data Acquisition System

This chapter provides different steps of hardware design of the proposed data acquisition system. The main element of the data acquisition system are sensors, microcontroller and communication devices.

3.1 Sensors

Table 3.1 provides the summary of different sensors and microcontrollers used.

Table 3.1: List of sensors and microcontroller used

Description	Part Number	Manufacturer
Ambient Temperature Sensor	LM35	National Semiconductor
Humidity Sensor	DHT11	Aosong Electronics
Barometric Pressure Sensor	BMP085	Bosch Sensortech
Light Intensity Sensor	TEMT6000	Sparkfun Electronics
Soil Hygrometer		ITEAD Studio
Real Time Clock	DS1307	Dallas Semiconductor
Microcontroller	ATmega328, AT91SAM3X8E	Atmel
Ethernet Module	ENC28J60	
Xbee Module		Digi International

3.1.1 Ambient Temperature Sensor

LM35 is precision integrated circuit temperature device with voltage linearly proportional to celcius temperature with a sensitivity of 10mv/$^{\circ}C$. LM35 is used for measurement of ambient temperature. LM35 don't require any additional calibration or trimming for the temperature output. The specification of LM35 is summarized in Table 3.2. The sensor has a wide range and draws a minimal current of 60 μA from the power supply. Fig. 3.1 shows the pin configuration of LM35 sensor.

Table 3.2: Specification of LM35

Operating Voltage	4 V to 30 V DC
Range	−55°C to 150°C
Accuracy	±5%, ±0.5°C

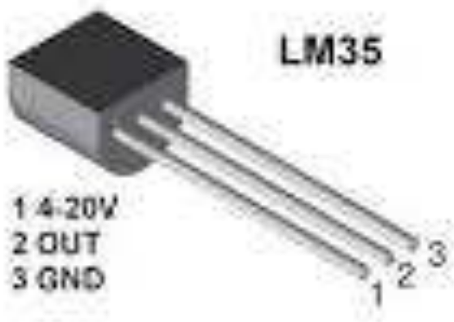

Figure 3.1: Pin configuration of LM35

3.1.2 Humidity Sensor

Humidity sensor (DHT11 or DHT22) includes a resistive-type humidity measurement component, an negative temperature coefficient (NTC) temperature measurement component and a high performance 8-bit microcontroller. The humidity sensor provides calibrated digital signal output. It has high reliability and excellent long-term stability. The specification of humidity sensor is summarized in Table 3.3. Fig. 3.2 shows the pin configuration of DHT22.

Table 3.3: Specification of DHT11

Power supply	3.3V - 5V DC
Output	4 pin single row
Range	20-90% RH
Accuracy	±5%, ±2°C
Resolution	1% RH, 1°C
Long term stability	< ±1% RH per year

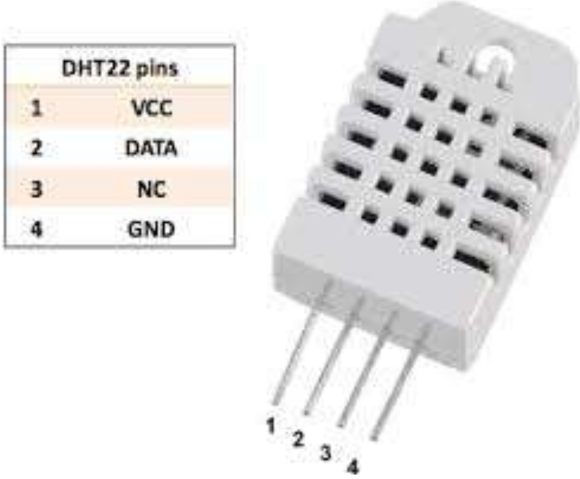

Figure 3.2: Pin configuration of DHT22

3.1.3 Barometric Pressure Sensor

BMP085 is an inter integrated circuit (I2C) protocol based barometric pressure sensor. The main advantage of I2C protocol is that the pressure sensor can be interfaced with the microcontrollers I2C pins. Serial data line (SDA) and serial clock line (SCL) are two open drain lines used for I2C communication. I2C is a multi-master, multi slave protocol. Therefore more than one sensor can be interfaced with one single pin. The specification of BMP085 barometric pressure sensor is summarized in Table 3.4. Fig. 3.3 shows the pin configuration of BMP085 sensor.

From barometric pressure, altitude can be calculated. The measured pressure is p and the pressure at sea level p_o is 1013.25 hPa. The altitude can be calculated using the following formula

$$Altitude = 44330\left(1 - \left(\frac{p}{p_o}\right)^{\frac{1}{5.255}}\right) \tag{3.1}$$

A pressure change Δp of 1 hPa corresponds to 8.43m at sea level.

Table 3.4: Specification of BMP085

Power supply	3V to 5V DC
Range	330 hPa to 1100 hPa
Resolution	0.03 hPa per 0.25 m

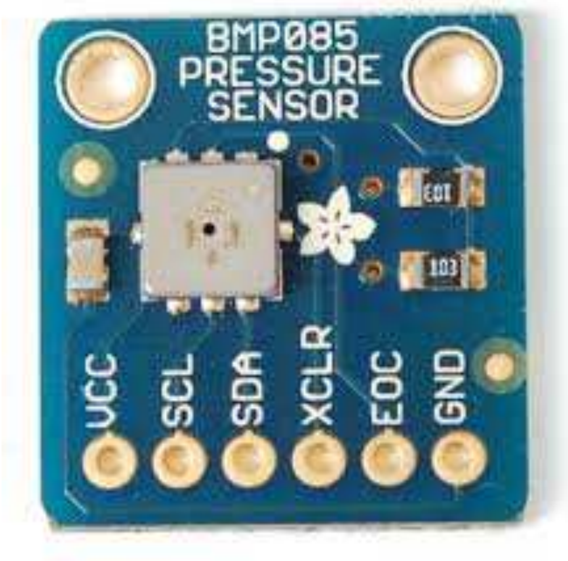

Figure 3.3: Pin configuration of BMP085

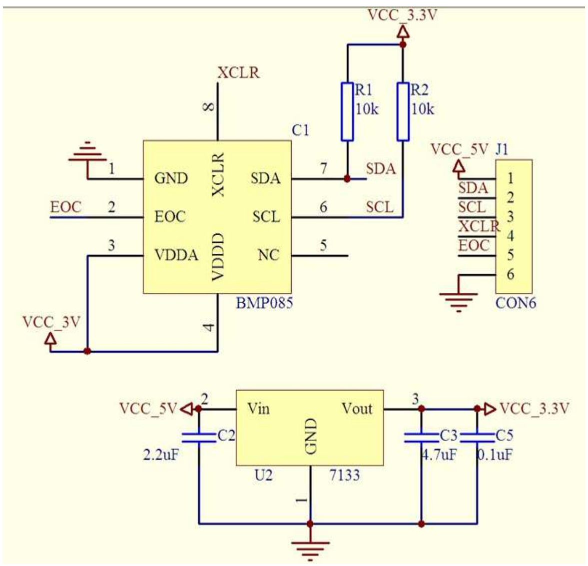

Figure 3.4: Circuit diagram of microcontroller and sensor interface

3.1.4 Ambient Light Sensor

To measure the light intensity in Lux, ambient light sensor (TEMT 6000) is used. TEMT 6000 is a silicon NPN epitaxial planar phototransistor in a miniature transparent mold for surface mounting onto a printed circuit board. The device is sensitive to the visible spectrum. Fig. 3.5 shows the pin configuration of TEMT6000 light intensity sensor.

Figure 3.5: Pin configuration of TEMT6000

3.1.5 Soil Hygrometer sensor

Soil moisture sensor is sensitive to ambient humidity and is generally used to detect the moisture content of soil. The operating voltage is 5V DC. The module provides output in both analog mode as well as digital mode. Fig. 3.6 illustrates the circuit diagram of soil hygrometer sensor. Soil hygrometer consists of a probe, which is inserted in to the soil to detect the moisture content of the soil. Fig. 3.7 shows the soil sensor.

3.1.6 Real Time Clock

Real time clock (RTC) is a computer clock which keeps track of current time. In data acquisition and data logging application the actual time of the acquired signal is required. Therefore, RTC is used to keep track of current time. DS1307 is an I2C protocol based RTC which is used in this application to keep track of current time. Real time clock consists of

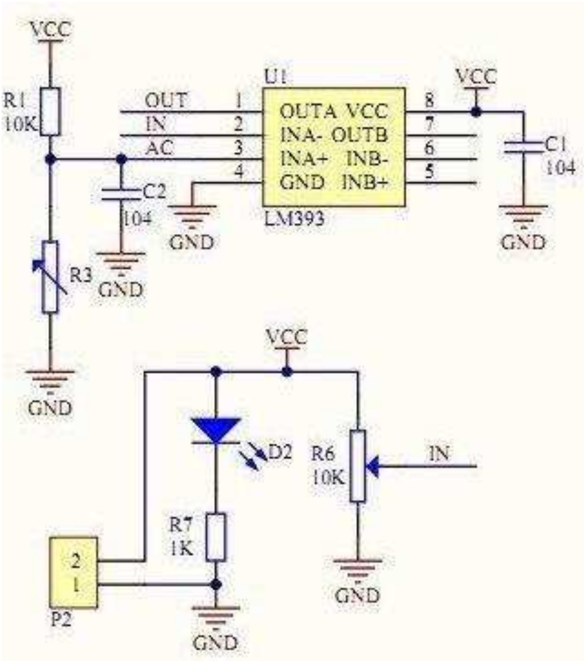

Figure 3.6: Circuit diagram of soil hygrometer sensor

Figure 3.7: Soil sensor

a crystal and a battery. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

There are other RTC as well such as DS3231. In DS3231, there is a temperature compensated oscillator.

Fig. 3.8 illustrates the circuit diagram of real time clock. Fig. 3.9 shows the RTC module.

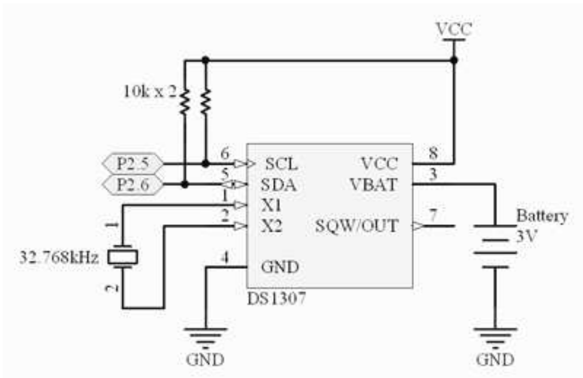

Figure 3.8: Circuit diagram of Real Time Clock

Figure 3.9: Real Time Clock

3.2 Microcontroller

3.2.1 ATmega328

This project implements multi channel data acquisition system using low cost microcontroller. One of the most widely used microcontroller unit is Arduino (i.e ATmega328) microcontroller. The features of the microcontroller is summarized in Table 3.5. The pinout of ATmega328 microcontroller is shown in Fig. 3.10. Fig. 3.12 presents the commercial Arduino UNO available in the market at a low price. ATmega328 microcontroller is popular because of inbuilt I2C and SPI protocol.

Table 3.5: Features of ATmega328

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage	7V to 12V
Input Voltage (Limits)	6V to 20V
Digital I/O pins	14
Analog Input Current	6
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for	3.3V Pin 50 mA
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

3.2.1.1 I2C Protocol

Inter integrated circuit (I2C) protocol was invented by NXP semiconductor (formerly known as Philips Semiconductor) in the year 1982. I2C protocol is used for connecting two or more peripherals with the CPU. I2C bus consists of two puns such as Serial Clock Line (SCL) and Serial Data Line (SDA). I2C is a multi master, multi slave, single ended, serial computer bus protocol. I2C has 7 to 10 bit address space and the communication speed is up to 3.4 Mbps. The schematic connection diagram of I2C bus is illustrated in Fig. 3.13. The working principle of I2C address is summarized as follows

1. Initialize I2C device with its address
2. Send READ request to that particular address

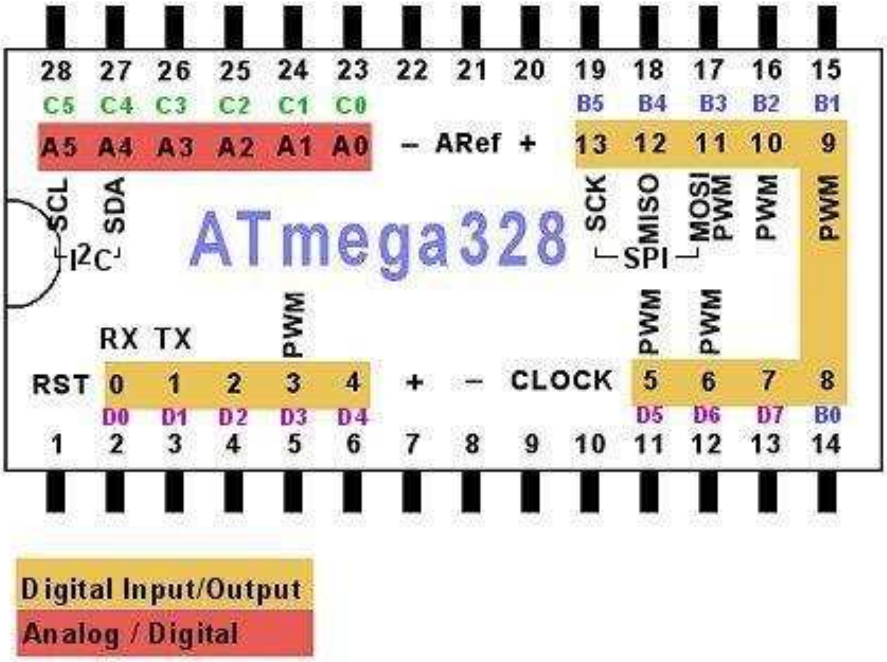

Figure 3.10: Pinout of ATmega328 microcontroller

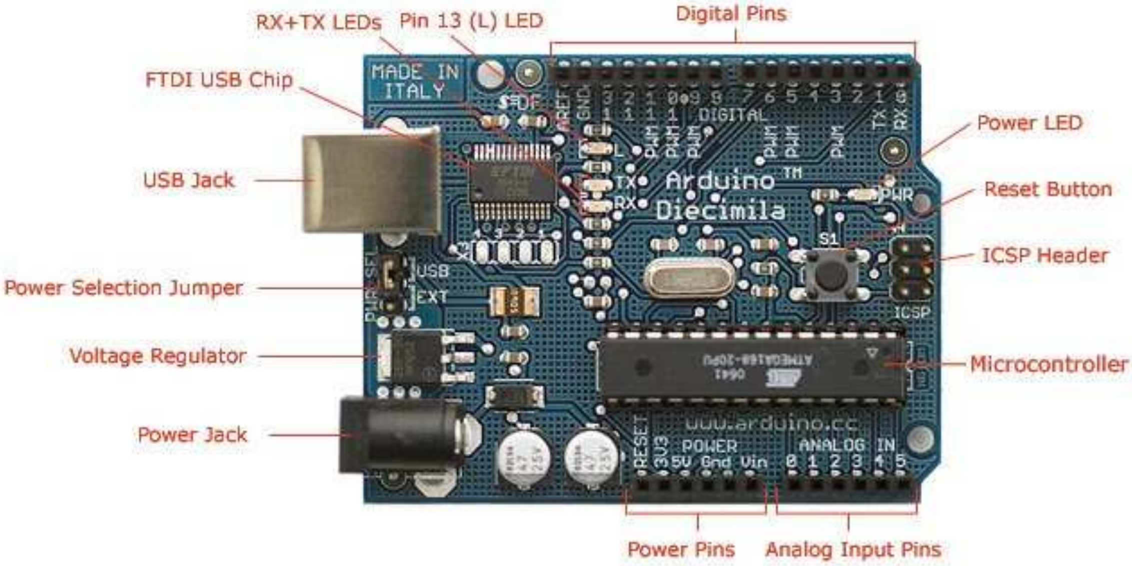

Figure 3.11: Arduino Uno

3. Decode the received reply to get the sensor data

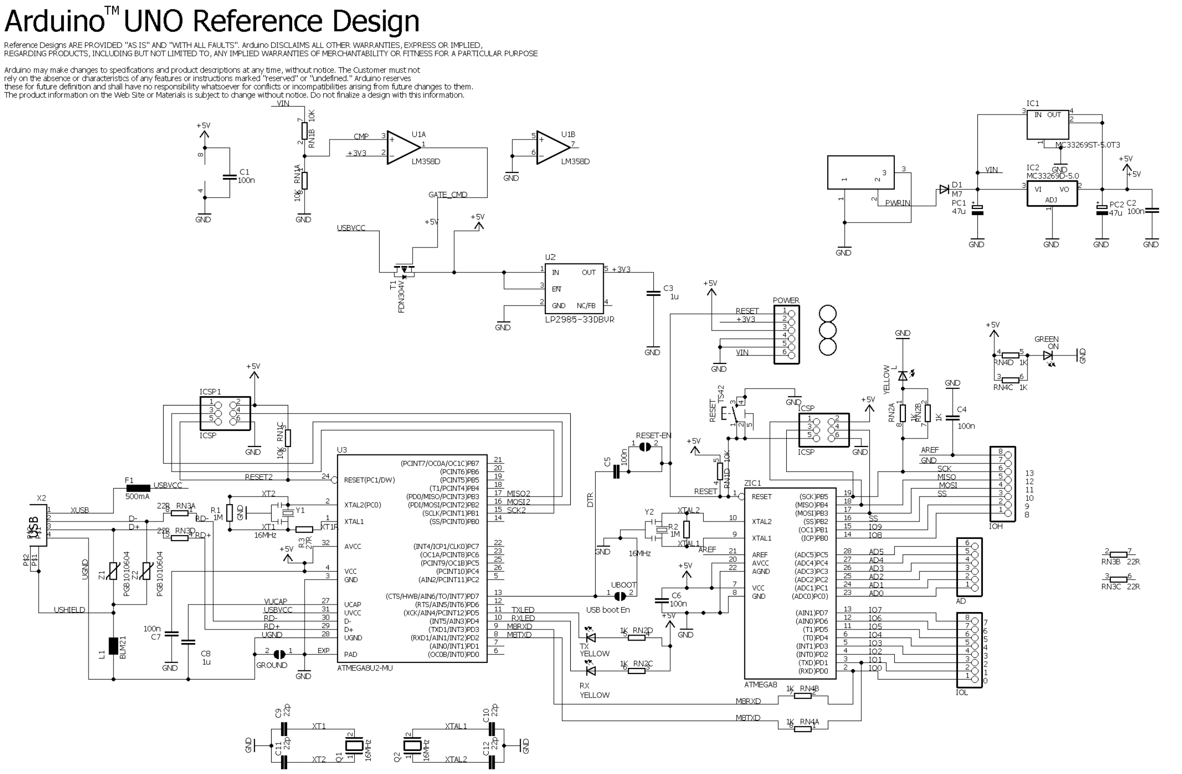

Figure 3.12: Schematic of ATmega328 microconroller

4. Read sensor data and log it

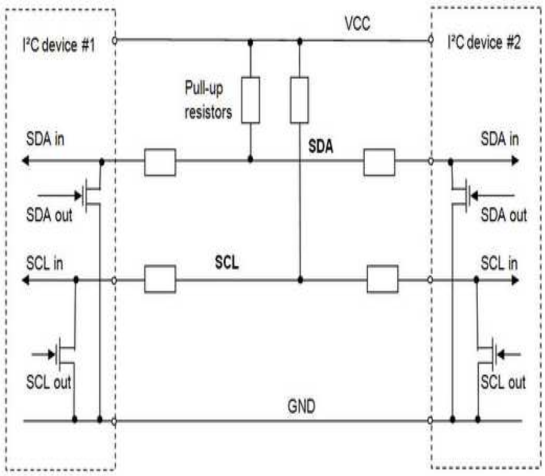

Figure 3.13: I2C protocol

3.2.1.2 Serial Peripheral Interface (SPI) Protocol

SPI is the acronym of Serial Peripheral Interface. SPI protocol has 3 signals such as clock signal SCLK, Master In Slave Out (MISO) and Master Out Slave In (MOSI). The operation of SPI protocol is summarized as follows.

1. SCLK, sent from the bus master to all slaves; all the SPI signals are synchronous to this clock signal;
2. A slave select signal for each slave used to select the slave to communicates with
3. A data line from the master to the slaves
4. A data line from the slaves to the master

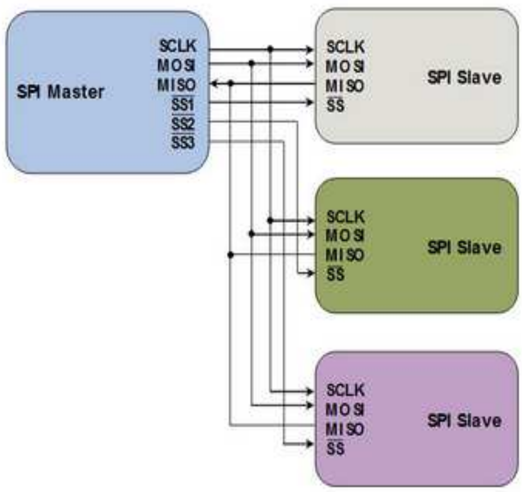

Figure 3.14: SPI protocol

3.2.2 AT91SAM3X8E

AT91SAM3X8E microcontroller widely known as Arduino Due is a 32 bit ARM Cortex M-3 CPU. The features of this microcontroller is summarized in Table 3.6. Arduino Due has in built 12 bit ADC and PWM resolution. Using a Micro-B USB cable, Arduino Due can be connected with a PC. Fig. 3.15 shows commercially available Arduino DUE (AT91SAM3X8E).

Table 3.6: Features of AT91SAMX8E

Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage (recommended)	7 to 12V
Input Voltage (limits)	6 to 16V
Digital I/O Pins	54
Analog Input Pins	12
Analog Outputs Pins	2 (DAC)
Total DC Output Current on all I/O lines	130 mA
DC Current for 3.3V Pin	800 mA
DC Current for 5V Pin	800 mA
Flash Memory	512 KB
SRAM	96 KB
Clock Speed	84 MHz

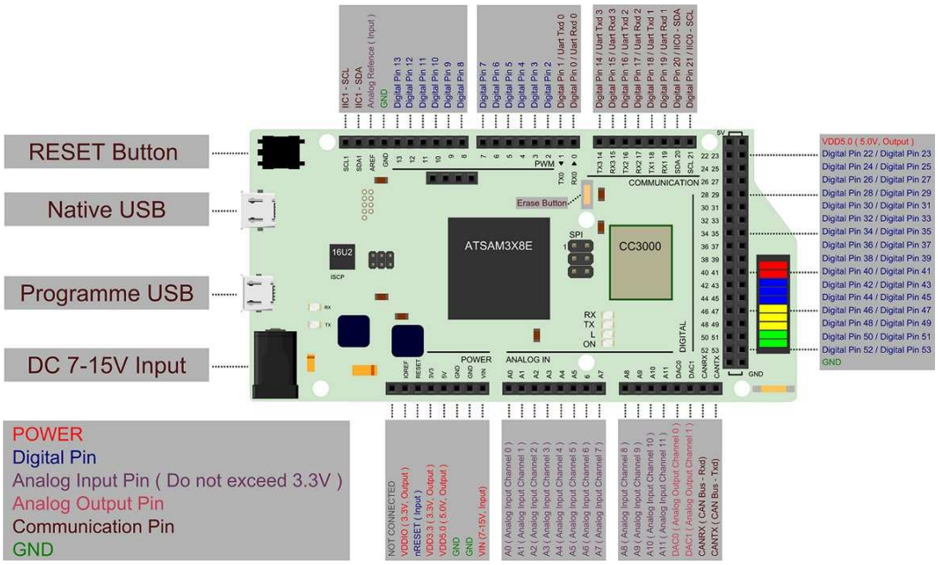

Figure 3.15: Arduino Due (AT91SAM3X8E)

3.3 Power Supply Unit

Power supply module is essential part of any project. Commercially available Arduino module comes with inbuilt USB port which provide reliable power supply. But in this project separate power supply module is developed. This project requires different voltage levels for different sensors and microcontroller. Table 3.7 summarizes the power requirement of different sensors and microcontroller units. Fig. 3.16 shows the circuit diagram of power supply used in this project which generates 5V and 3.3V DC. To generate 5V DC from 240V, 50Hz AC signal, step down transformer, diode bridge rectifier (1N4007) and voltage regulator IC (LM7805) is used. The current rating of LM7805 is 1.5A and there is internal current limiting and thermal shut down features available in these kind of regulators. These features make these regulators immune of thermal and current overload. The current capacity of the power module should also be taken in to consideration. This power module can provide up to 700mA of current and it is enough for safe working of all the sensors and microcontrollers. Per I/O pin the DC current requirement is 40mA. DC current for supply and ground is 200mA. GND pin has current sink of 400mA.

To generate 3.3V DC from 5V supply, a 3.3V zener diode is used. Smoothing capacitors are used to reduce the ripple from the output DC voltage. Apart from using 3.3V zener diode, LM1117 low dropout linear regulator can also be used to generate 3.3V DC voltage. Fig. 3.17 provides implementation of 3.3V DC using a different circuit. Fig. 3.18 presents the hardware design of 5V supply.

Table 3.7: Power Supply Unit

Module	Voltage Level
LM35	5V
DHT11	5V
BMP085	3.3V
TEMT6000	5V
Soil Hygrometer	5V
DS1307	5V
ATmega328	5V
AT91SAM3X8E	3.3V
ENC28J60 module	3.3V
Xbee module	5V

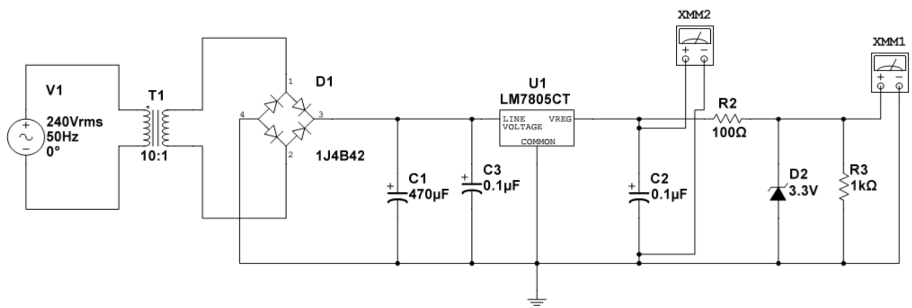

Figure 3.16: Power Supply Unit

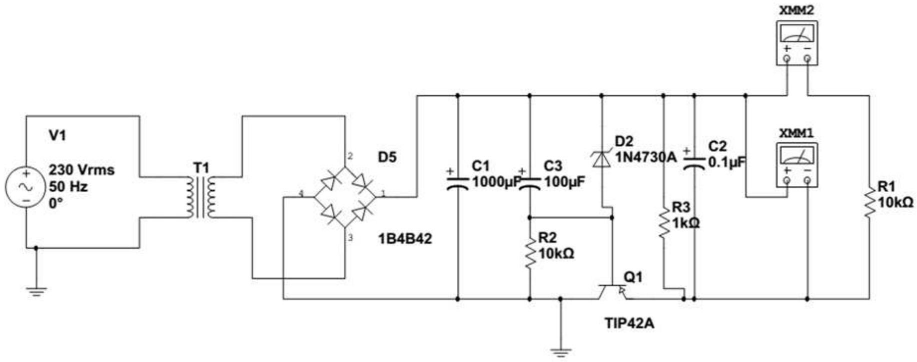

Figure 3.17: 3.3V supply

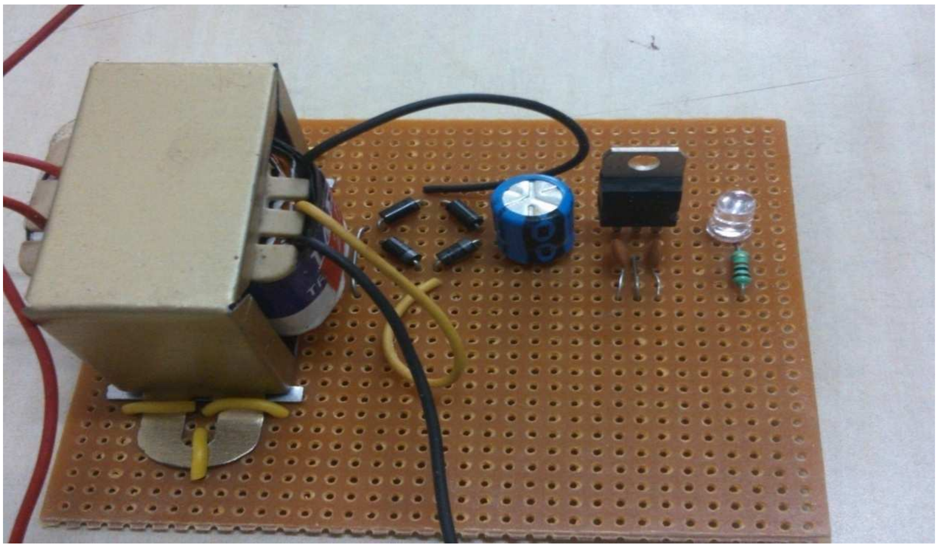

Figure 3.18: Hardware implementation of 5V supply

3.4 Software Implementation

3.4.1 Programming ATmega328

For programming the ATmega328 microcontroller, the TX and RX pins of the microcontroller are used. Using USB FTDI interface the microcontroller code (written in C++ language) is downloaded in the microcontroller. Fig. 3.19 presents the programming of the microcontroller. Fig. 3.20 shows the sketch loader used for microcontroller programming.

The driving force behind the PC based DAQ system is the software. Programming a DAQ system is accomplished by the following three ways

- Hardware level programming

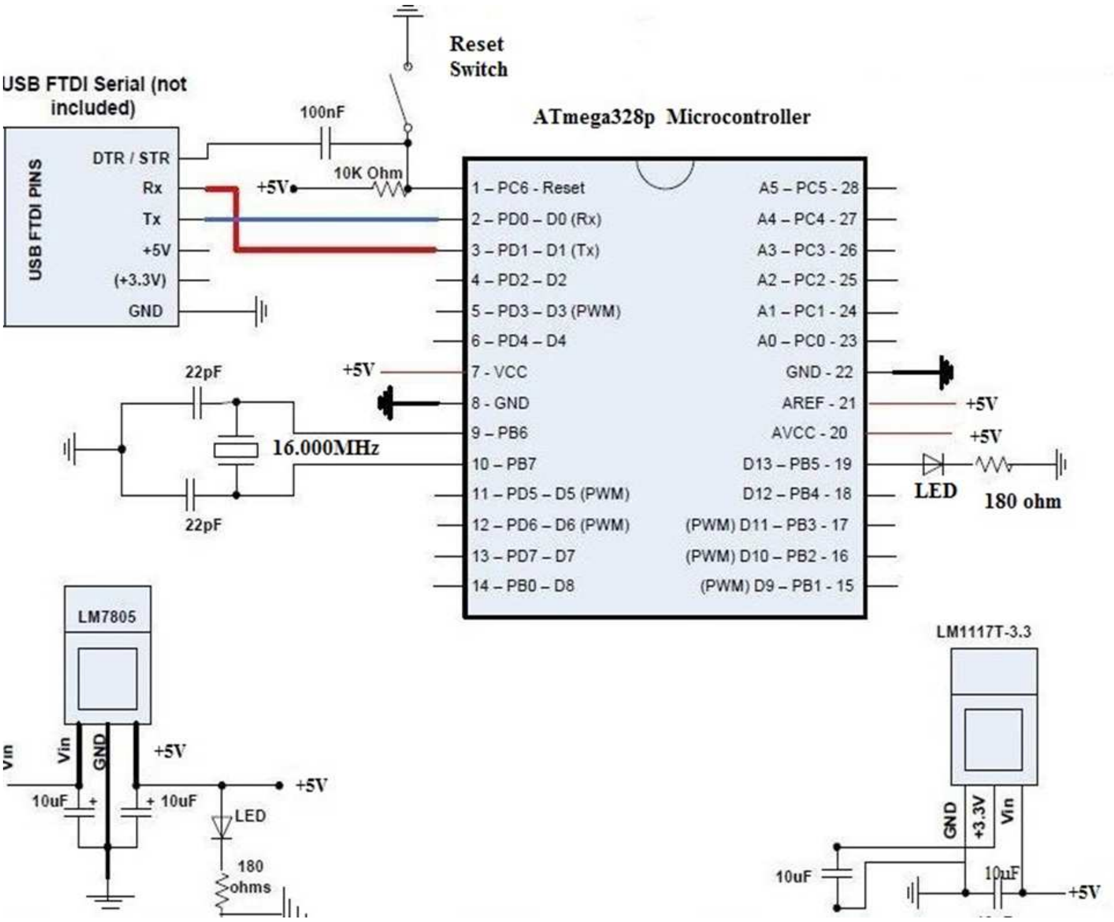

Figure 3.19: Programming ATmega328

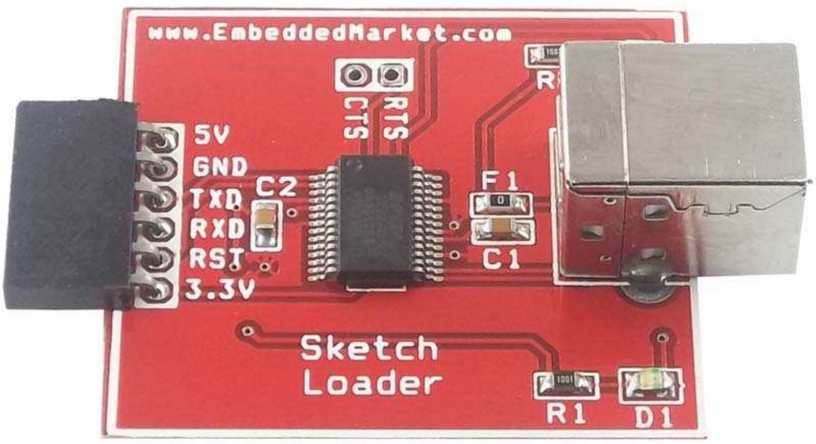

Figure 3.20: Sketch loader

- Driver level programming
- Package level programming

3.4.1.1 Hardware level programming

In hardware level programming data acquisition hardwares registers are directly programmed. A programming language that offers low level bit function like c and assembly language are used to enter the code values into the resistor. This of programming is complex and takes a lot of time.

3.4.1.2 Driver level programming

In this type of programming high level functions from libraries provided by the card manufacturer are simply called. This function complete tasks such as A/D conversion, into a single call. High level programming language like c/c++, Pascal and BASIC can be used to call the library functions.

3.4.1.3 Package level programming

In package level programming, the application program performs the hardware level function. Software packages integrate data analysis, presentation and instrument control capabilities into a single interface. This programming offers data logging, data analysis and real time graphic display.

3.5 Serial Communication Based Data Acquisition System

This section provides the results of serial communication based data acquisition system. The sensor and the microcontroller unit are connected with the COM port of PC. The block diagram of serial communication based data acquisition system is shown in Fig. 3.21. The flow chart of serial communication based DAQ is shown in Fig. 3.22. In this serial communication, the sensor data are read using the input port of microcontroller and microcontroller converts the input signal to relevant numeric value. This procedure is repeated time and again. In this project, there are different sensor interfaces such as analog output, digital output and I2C output.

The data transmitted from serial port of microcontroller is interfaced with NI-LabVIEW using VISA protocol. VISA stands for Virtual Instrument Software Architecture developed

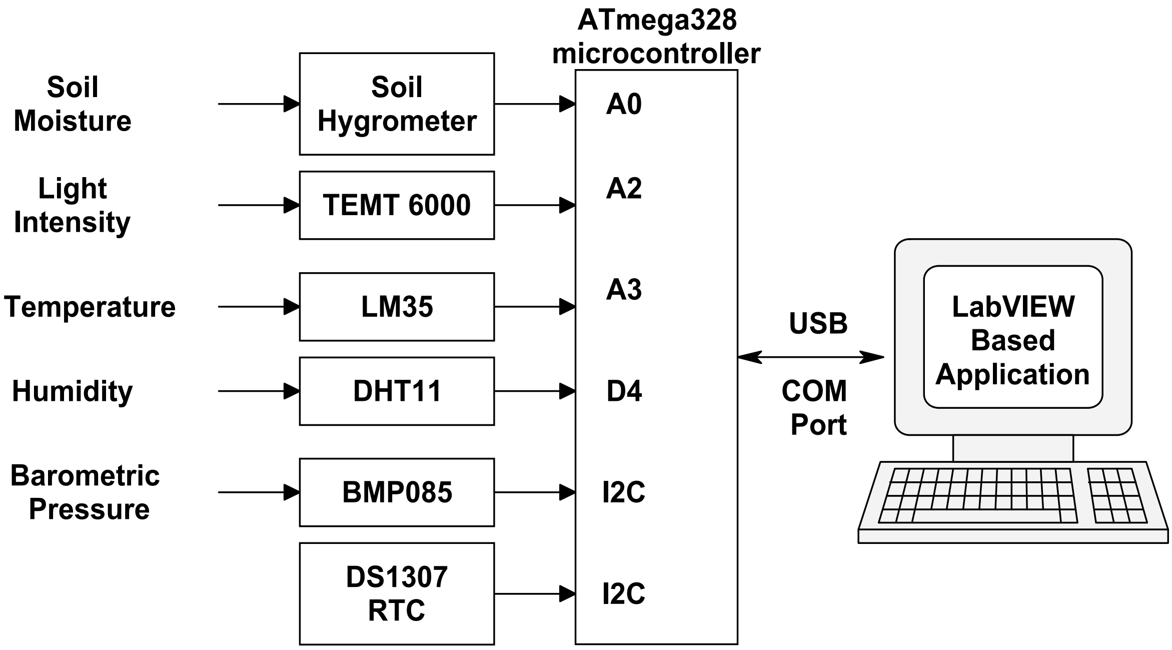

Figure 3.21: Block diagram representation of the proposed data acquisition system

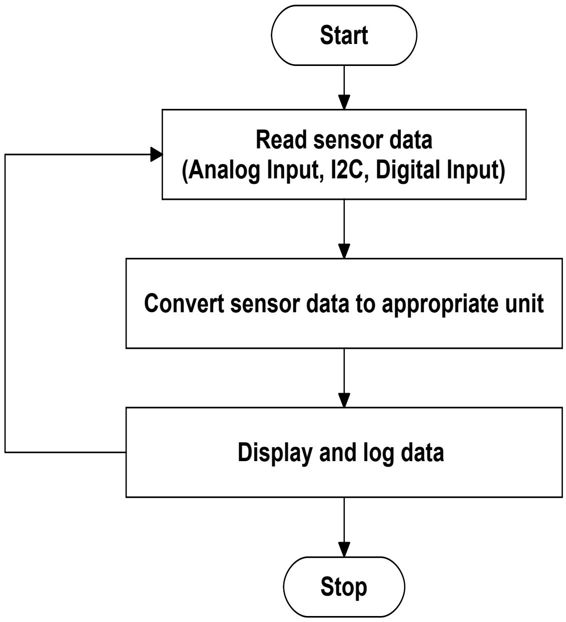

Figure 3.22: Flow chart of serial communication based DAQ

by VXI Plug and Play Systems Alliance. It is a standard for configuring, programming, and troubleshooting instrumentation systems comprising GPIB, VXI, PXI, Serial, Ethernet, and/or USB interfaces. VISA provides the programming interface between the hardware and development environments. The basic operations of VISA are Establish link with VISA driver Open a communication channel Communicate with the instrument Terminate the session Close the link with VISA driver The microcontroller communicates via its USB port to the PC but this is presented to the LabVIEW as a COM port. VISA collects the data from the COM port and displays the data in the LabVIEW front panel.

3.6 Ethernet Module (ENC28J60)

To facilitate ethernet communication in DAQ, ENC28J60 module is used. This is a IEEE 802.3 compatible ethernet module. This module is connected to SPI port of the microcontroller. Ethernet module consists of 25MHz clock, ENC28J60 SPI chip and HR911102A RJ45 socket. This module supports half duplex and full duplex communication mode. It supports different TCP/IP protocol stack such as IPv4, UDP, TCP, DHCP, ICMP, FTP and HTTP. ENC28J60 chip is a 28 pin DIP, 10 BASE-T stand alone controller with on board MAC and PHY. It has 8KB buffer RAM and SPI protocol.

The pin connection of microcontroller and ethernet module is described below

1. ENC SO to Pin 12 of Arduino
2. ENC SI to Pin 11 of Arduino
3. ENC SCK to Pin 13 of Arduino
4. ENC CS to Pin 8 of Arduino
5. ENC VCC to 3.3V
6. ENC GND to GND

The circuit diagram of ethernet module is presented in Fig. 3.23. The working principle of ethernet based DAQ is summarized in Fig. 3.25. For linkup of measurement values to web server, a webserver has to setup with IP address, subnet mask and port ID. An appropriate IP address is considered. Ethernet communication is handshake based communication and if any request is received then the data is displayed. If the request is not received, then the web server don't display the data. The working principle of ethernet based DAQ is summarized in Fig. 3.25. Fig. 3.26 shows the connection diagram of ethernet module with the microcontroller.

The specifications of ENC28J60 is summarized below.

1. MAC address and Physical address available
2. TX/RX RAM buffer is 8192 bytes
3. Interrupt Pin is 1
4. Operating voltage is 3.3V
5. Temperature range is -40°C to 85°C
6. Maximum speed is 25MHz
7. 10Base-T stand alone ethernet controller
8. Programmable padding and CRC generation
9. SPI interface speed up to 10 Mbps
10. Programmable automatic rejection of erroneous packets
11. Programmable automatic retransmission on rejection
12. Consists of internal DMA module
13. Fast data throughput and hardware assisted IP checksum calculations

3.7 Ethernet Based Data Acquisition System

The block diagram of ethernet based data acquisition system is shown in Fig. 3.27. Using ethernet based data acquisition system, weather data can be accessed using PC and cellular device with a network connectivity.

3.8 Xbee Module

To facilitate wireless communication in multi channel data acquisition system, Xbee module is used. The salient features of Xbee module is summarized in Table 3.8. The block diagram of wireless module is shown in Fig. 3.28 respectively.

The block diagram of wireless based data acquisition system is shown in Fig. 3.29.

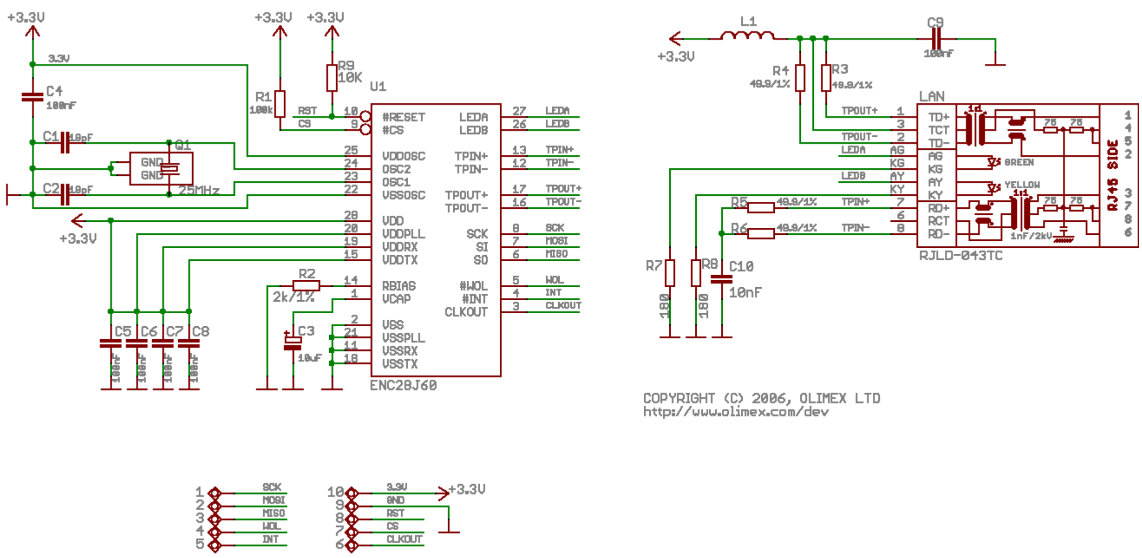

Figure 3.23: Circuit diagram of ethernet module

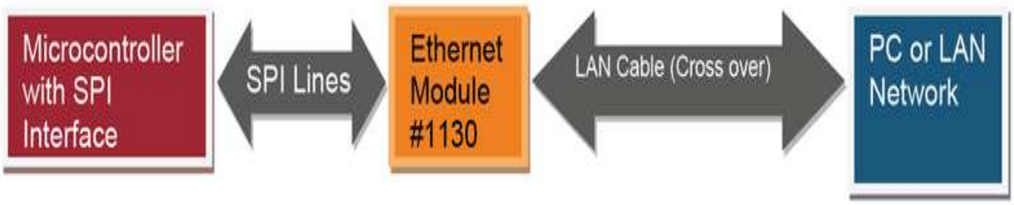

Figure 3.24: Block diagram of ethernet communication

Table 3.8: Xbee Module

Xbee Module	IEEE 802.15.4 standard
Radio	Direct Sequence Spread Spectrum
Frequency	2.4GHz
Indoor range	30 m
Outdoor range	90 m
Transmitter power	1 mW (0 dBm)
Receiver sensitivity	-92 dBm
Maximum data rate	2 Mbps

3.9 Cost Analysis

The cost of entire project is summarized in Table 3.9. The total cost of the entire design and real time implementation of the project is INR 6500. This is an economical design and

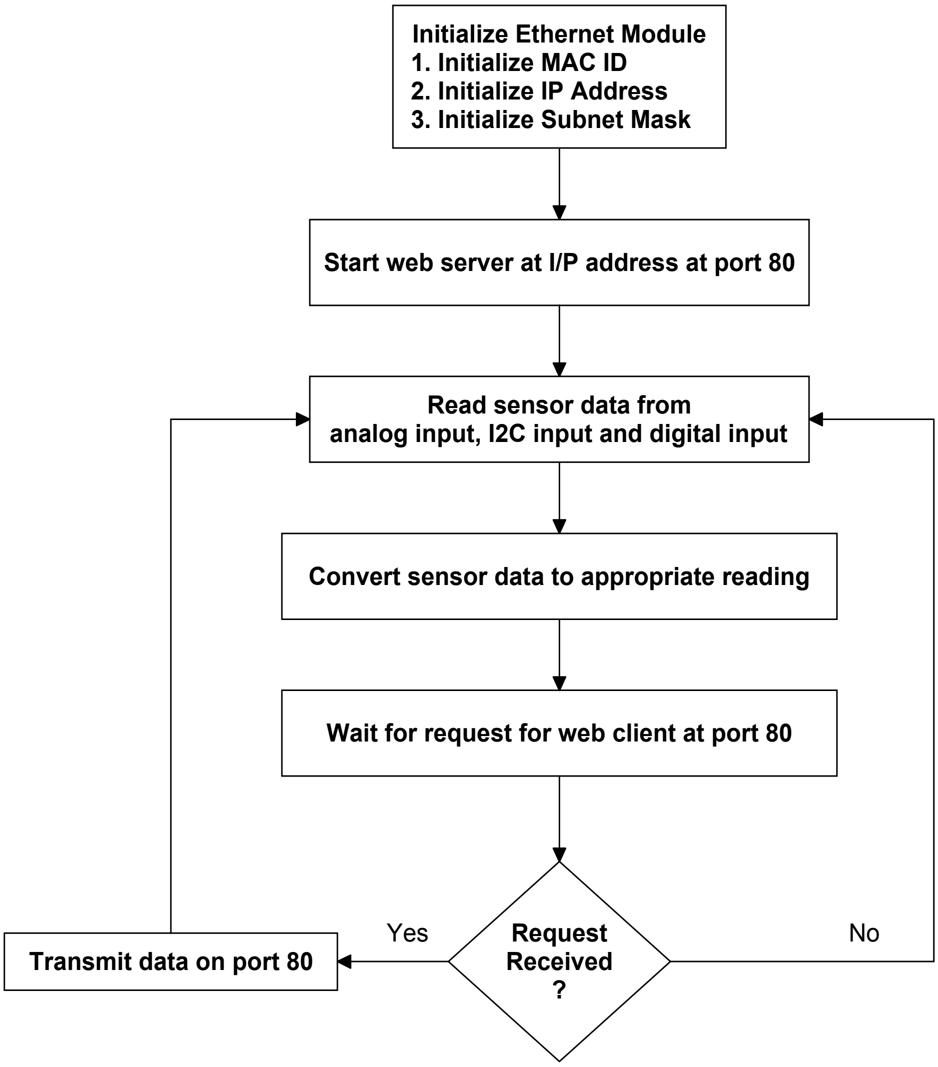

Figure 3.25: Flow chart of ethernet based DAQ

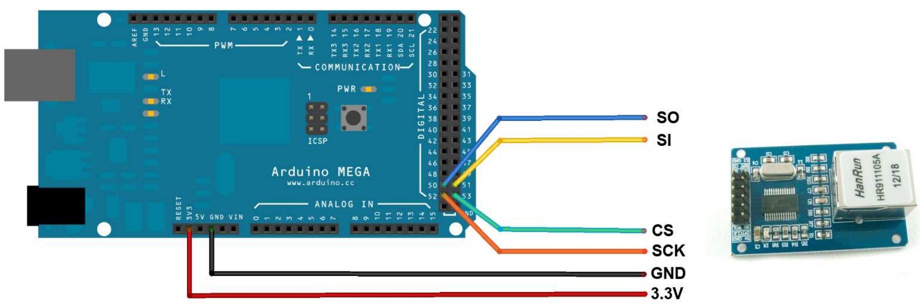

Figure 3.26: Connection diagram of ethernet communication

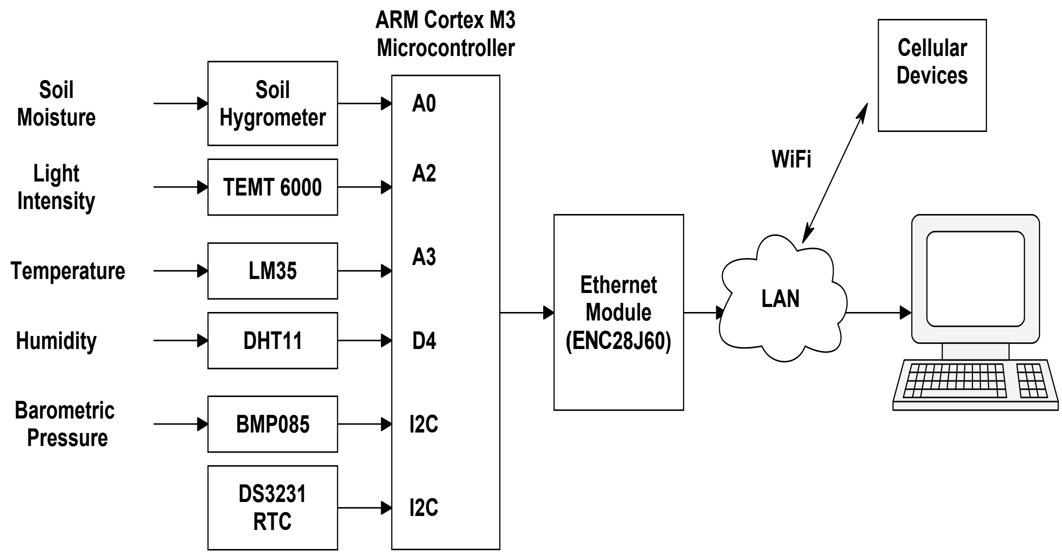

Figure 3.27: Block diagram of ethernet based data acquisition system

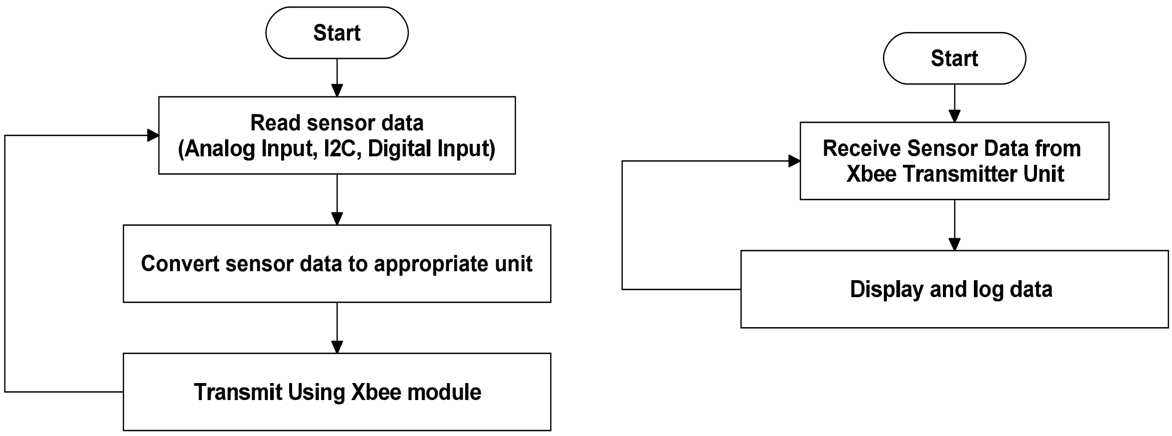

Figure 3.28: Flow chart of wireless transmission based data acquisition system

the design can be optimized further.

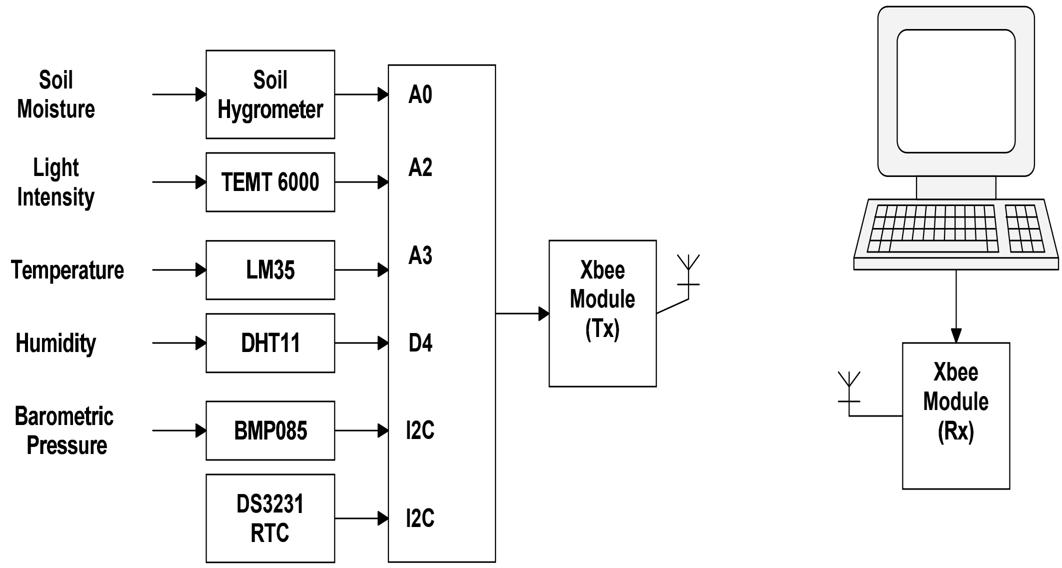

Figure 3.29: Block diagram of wireless based data acquisition system

Table 3.9: Bill of materials

Description	Part Number	Cost (INR)
Ambient Temperature Sensor	LM35	46
Humidity Sensor	DHT11	149
Barometric Pressure Sensor	BMP085	339
Light Intensity Sensor	TEMT6000	120
Soil Hygrometer		140
Real Time Clock	DS1307	123
Microcontroller	ATmega328, AT91SAM3X8E	2650
Ethernet Module	ENC28J60	280
2mW Xbee Antenna		2000
Xbee Adapter		245
Xbee Shield		350
Transformer	12V, 750 mA	50
Voltage Regulator	LM7805	10

CHAPTER 4

Results and Discussion

4.1 Serial Communication

In serial communication, all the sensors are connected to the input port of microcontroller and the microcontroller reads the sensor data at a specified interval and the data is shown in serial port of microcontroller and COM port of PC. The baud rate of serial communication is 9600.

Fig. 4.1 and 4.2 presents the hardware developed for data acquisition and data logging application. The data of microcontroller is sent to LabVIEW via serial port using VISA protocol. A LabVIEW application is developed using VISA. The front panel of the above application is shown in Fig. 4.3. The LabVIEW application has data storage capacity and a graphic user interface. Fig. 4.4 illustrates the sensors data displayed at the serial port of PC.

4.2 Ethernet Communication

Using ethernet based data acquisition system, the weather data can be accessed in web browser as shown in Fig. 4.5. The IP address gateway is 192.168.48.1, Subnet mask is 255.255.252.0, IP address DNS server is 192.168.1.250 and the IP address is 192.168.44.244

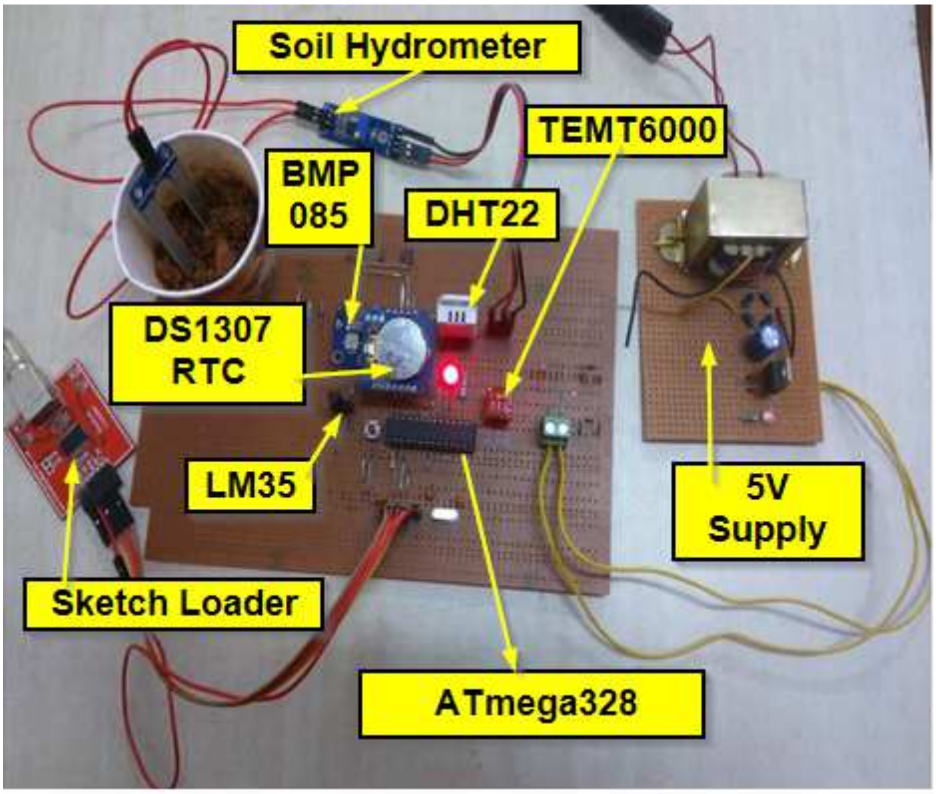

Figure 4.1: Hardware implementation of PC based multi channel data acquisition system

4.3 Wireless Communication

Using wireless data acquisition system, the weather data can be accessed in COM port of PC as shown in Fig. 4.6. The data can be transmitted up to 100m using Xbee transmitter and Xbee receiver.

4.4 Real Time Testing of Data Acquisition System

The Arduino Due microcontroller is interfaced with a 2 GB MicroSD card through the SPI (Serial Peripheral Interface) bus using three pins(MISO, MOSI and SCK). The measured parameters (Time in Second, Temperature in Celsius, Pressure in Pascal, Humidity as %RH and Luminous Intensity as Lux) are written to the SD card as Comma Separated Values (CSV) in a Text file. A new measurement is taken at every second and is appended to the text file. This system is left for about 24 Hours to be able to record all the parameters

Figure 4.2: Complete hardware setup of PC based multi channel data acquisition system

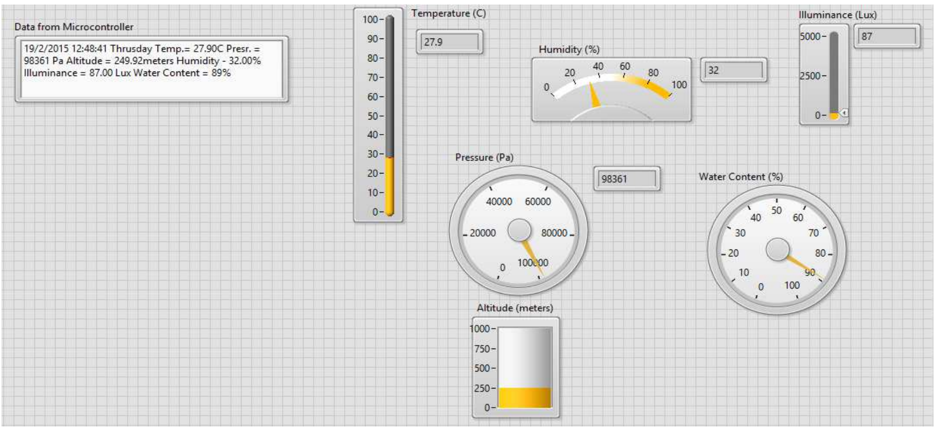

Figure 4.3: LabVIEW front panel for data logging system

changed throughout the day. After a certain time i.e. 24 hours the SD card is removed from the microcontroller circuit. The data is then imported into a computer via a USB card reader. A python script is written which reads the CSV text file and separates the data into different arrays which contain the different measured parameters. These arrays are then plotted using MatPlotLib library. Fig. 4.7 shows the temperature profile of 24 hour. Fig.

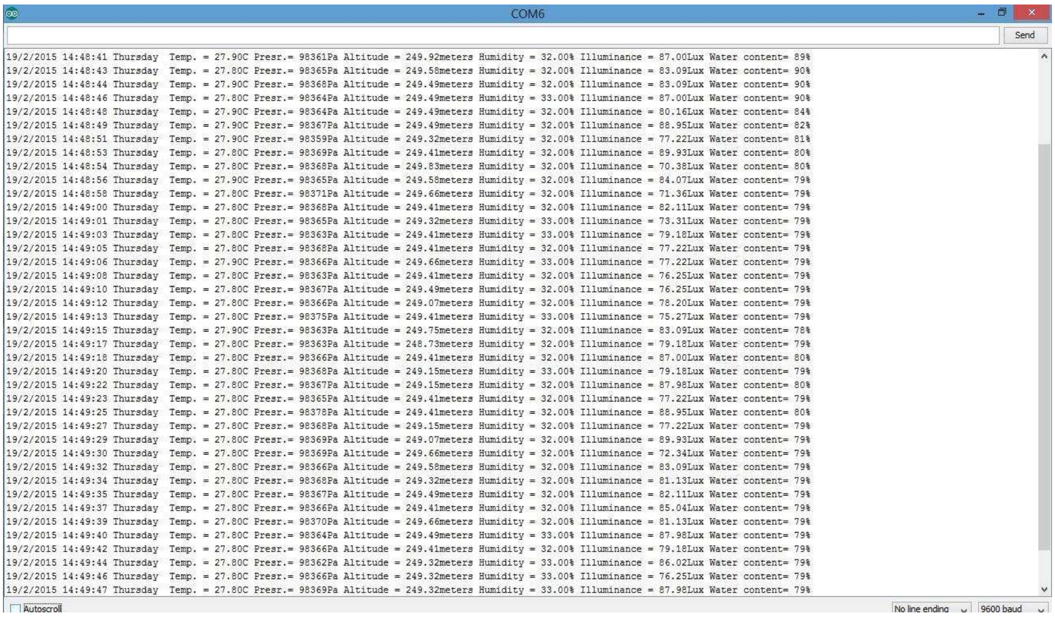

Figure 4.4: Sensors acquired data displayed on PC serial port

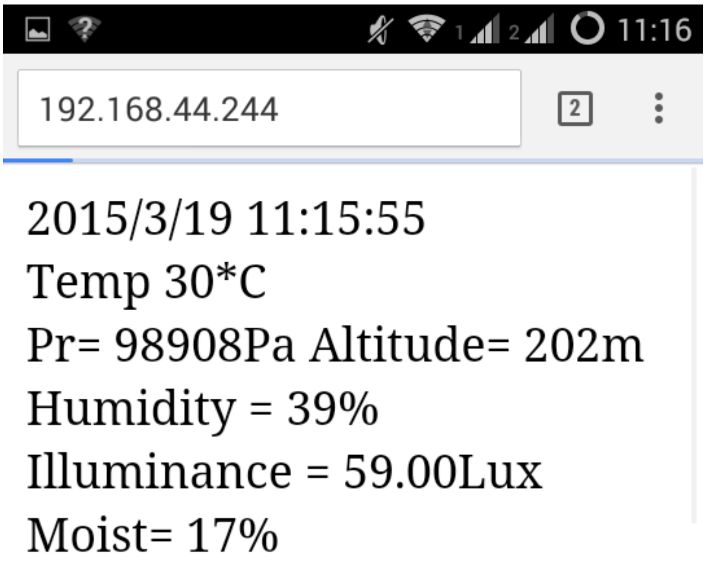

Figure 4.5: Weather information displayed in web browser

4.8 shows the pressure profile of 24 hour. Fig. 4.9 shows the humidity profile of 24 hour. Fig. 4.10 shows the light intensity profile of 24 hour.

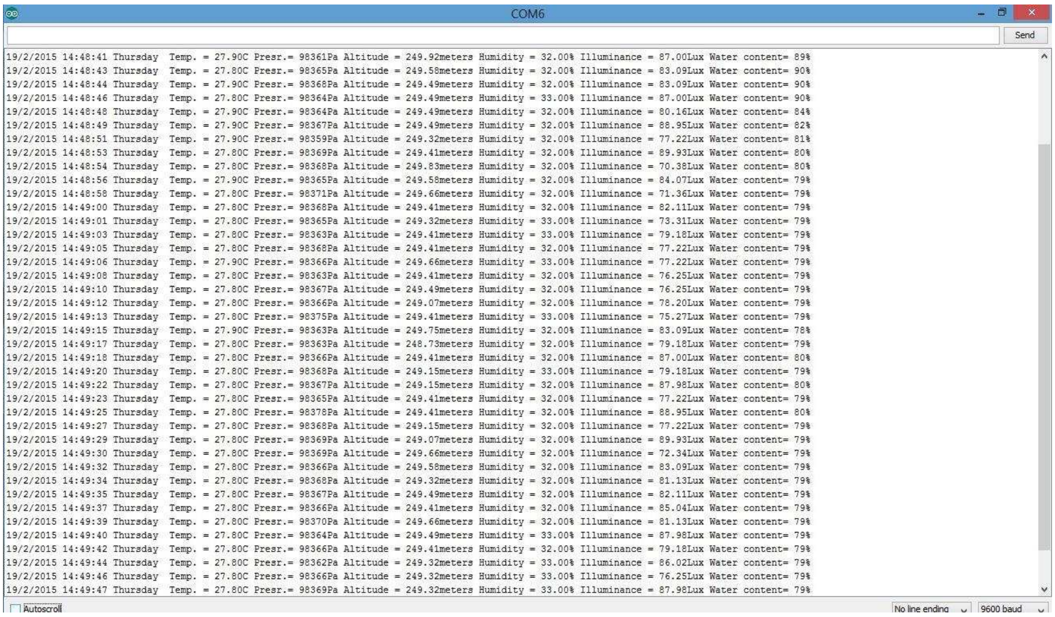

Figure 4.6: Sensors acquired data displayed on PC COM port in wireless DAQ

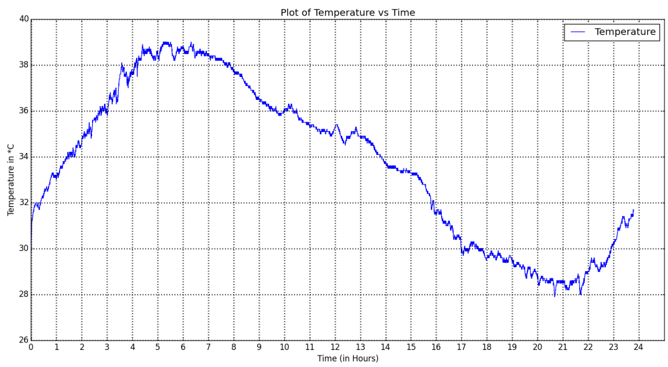

Figure 4.7: Temperature plot of 24 hours

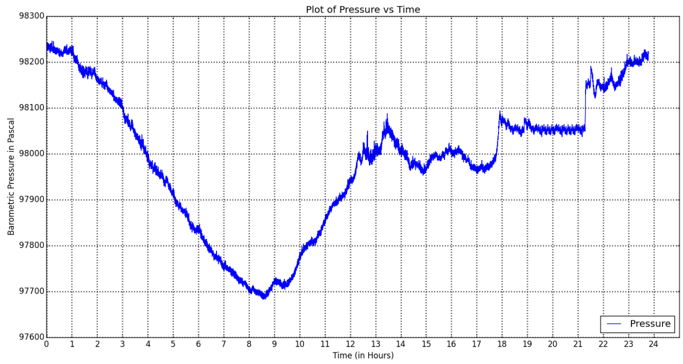

Figure 4.8: Pressure plot of 24 hours

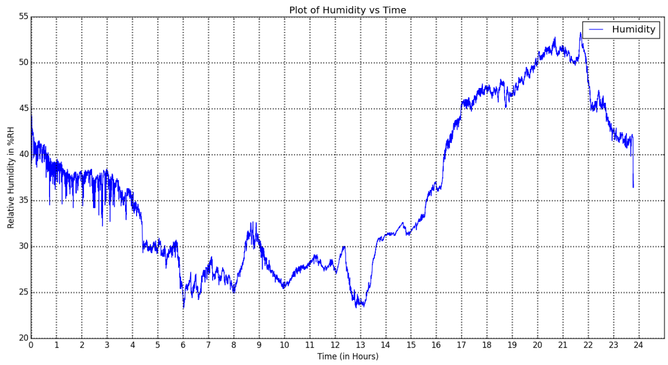

Figure 4.9: Humidity plot of 24 hours

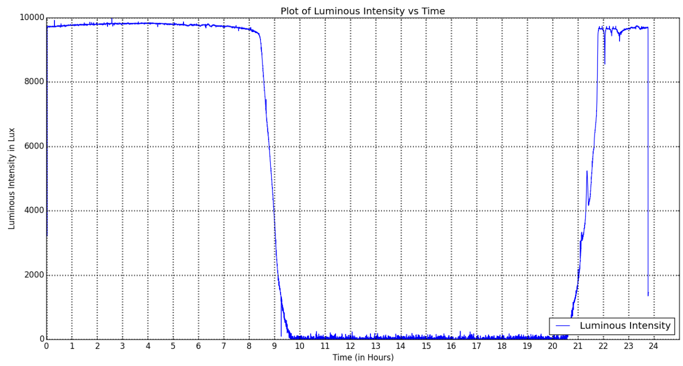

Figure 4.10: Light intensity plot for 24 hours

CHAPTER 5

Conclusion

This thesis discusses the design of a low cost multi channel data acquisition and data logging system for meteorological application. Different physical parameters such as ambient temperature, barometric pressure, altitude, humidity, moisture content and illuminance are acquired from environment using low cost microcontroller. Three different communication interfaces such as serial communication, wireless communication and ethernet communication are used in the multi channel data acquisition system.

In serial communication, the acquired data is displayed in the COM port of PC with an appropriate time stamp. In wireless communication, Xbee module is used to transmit data up to a certain distance. In ethernet based communication, ethernet module consists of ENC28J60 IC is used for uploading the weather data in ethernet. By this method, the user can access the weather information from the web browser from a LAN network.

The data acquisition system with different communication interface is tested in real environment for 1 day (i.e 24 hours) in April 2015. Xbee module is connected and the data is simultaneously plotted from COM port of PC. The real time trend of ambient temperature, relative humidity, barometric pressure and light intensity (lux) data are plotted using MatPlotLib of Python programming language.

5.1 Salient Features of Developed System

- The developed weather station with five different sensors, wireless communication and ethernet communication is of low cost i.e INR 6500. The cost is lower than commercially available weather station platform with same specifications.
- The developed system comprises of open source hardware and open source software programming (python is used for real time plotting). The developed system don't use any proprietary hardware or software.

5.2 Future Scope

- More sensors such as wind speed sensor, wind direction sensor, rain fall detection sensor can be used for acquiring more environmental parameters.
- Statistical analysis of past and present measurement values can be carried out to predict the future values
- Multiple number of wireless nodes can be setup at different strategic locations and all the sensor nodes can be connected to a central hub such as weather condition of different location can be summarized.
- More informative web application can be built to provide more useful information to the user.

Bibliography

[1] S. Ameur, M. Laghrouche, and A. Adane, "Monitoring a greenhouse using a microcontroller-based meteorological data-acquisition system," *Renewable Energy*, vol. 24, no. 1, pp. 19–30, 2001.

[2] Z. Su, J. Huschle, S. Redfield, T. Qiao, H. Liu, and F. Liu, "High-speed real-time multi-channel data-acquisition unit: Challenges and results," in *Proc. IEEE 11th Consumer Communications and Networking Conference (CCNC)*, 2014, pp. 105–112.

[3] H. Wang, X. Nie, D. Li *et al.*, "Design of pic microcontroller-based high-capacity multi-channel data acquisition module," in *Proc. 2012 International Conference onMeasurement, Information and Control (MIC)*, vol. 2, 2012, pp. 685–688.

[4] S. Rosiek and F. Batlles, "A microcontroller-based data-acquisition system for meteorological station monitoring," *Energy Conversion and Management*, vol. 49, no. 12, pp. 3746–3754, 2008.

[5] M. Benghanem, "A low cost wireless data acquisition system for weather station monitoring," *Renewable Energy*, vol. 35, no. 4, pp. 862–872, 2010.

[6] F. Shariff, N. A. Rahim, and W. P. Hew, "Zigbee-based data acquisition system for online monitoring of grid-connected photovoltaic system," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1730–1742, 2015.

[7] M. Fuentes, M. Vivar, J. Burgos, J. Aguilera, and J. Vacas, "Design of an accurate, low-cost autonomous data logger for pv system monitoring using arduino that complies with iec standards," *Solar Energy Materials and Solar Cells*, vol. 130, pp. 529–543, 2014.

[8] M. H. Waseem, M. Alamzeb, B. Mustafa, F. Malik, M. Shakir, and M. A. Khan, "Design of a low-cost underwater wireless sensor network for water quality monitoring," *IETE Journal of Research*, vol. 59, no. 5, pp. 523–534, 2013.

[9] D. S. Simbeye, J. Zhao, and S. Yang, "Design and deployment of wireless sensor networks for aquaculture monitoring and control based on virtual instruments," *Computers and Electronics in Agriculture*, vol. 102, pp. 31–42, 2014.

[10] N. Wang, N. Zhang, and M. Wang, "Wireless sensors in agriculture and food industryrecent development and future perspective," *Computers and electronics in agriculture*, vol. 50, no. 1, pp. 1–14, 2006.

[11] F. Touati, M. Al-Hitmi, K. Benhmed, and R. Tabish, "A fuzzy logic based irrigation system enhanced with wireless data logging applied to the state of qatar," *Computers and Electronics in Agriculture*, vol. 98, pp. 233–241, 2013.

[12] R. Ionel, G. Vasiu, and S. Mischie, "Gprs based data acquisition and analysis system with mobile phone control," *Measurement*, vol. 45, no. 6, pp. 1462–1470, 2012.

[13] W.-S. Jang, W. M. Healy, and M. J. Skibniewski, "Wireless sensor networks as part of a web-based building environmental monitoring system," *Automation in Construction*, vol. 17, no. 6, pp. 729–736, 2008.

[14] M. Benghanem, "Measurement of meteorological data based on wireless data acquisition system monitoring," *Applied energy*, vol. 86, no. 12, pp. 2651–2660, 2009.

[15] B. Rajesh Kumar, K. Sridharan, and K. Srinivasan, "The design and development of a web-based data acquisition system," *IEEE Transactions on Instrumentation and Measurement*, vol. 51, no. 3, pp. 427–432, 2002.

[16] I. Morsi and L. M. El-Din, "Scada system for oil refinery control," *Measurement*, vol. 47, pp. 5–13, 2014.

[17] F. F. Martín, M. V. Llopis, J. C. Rodríguez, J. B. González, and J. M. Blanco, "Low-cost open-source multifunction data acquisition system for accurate measurements," *Measurement*, vol. 55, pp. 265–271, 2014.

[18] S. Padhee and Y. Singh, "Data logging and supervisory control of process using labview," in *Proc. IEEE Students' Technology Symposium (TechSym)*, 2011, pp. 329–334.

[19] L. Lixin and D. Diankuan, "Design of multi-channel high-precision data acquisition system based on ad574a," in *Proc. 2011 Second International Conference on Mechanic Automation and Control Engineering (MACE)*, 2011, pp. 1199–1201.

[20] S. Sreelal, S. Jose, B. Varghese, M. Suma, R. Sumathi, M. Lakshminarayana, and M. Nayak, "Data acquisition and processing at ocean bottom for a tsunami warning system," *Measurement*, vol. 47, pp. 475–482, 2014.

Publications

Conference Proceedings

- **Nisha Kashyap** and Umesh Chandra Pati, "Multi channel data acquisition and data logging system for meteorology application," *Proc. IEEE International Conference on ICSTM*, 2015, pp. 237-242.

National Journal

- **Nisha Kashyap** and Umesh Chandra Pati, "Hardware implementation of PC based DAQ," Journal of Instrument Society of India (To be communicated)

Serial Communication

```
#include <Wire.h>
#include "RTClib.h"
#include <Adafruit_BMP085.h>
RTC_DS1307 RTC;
Adafruit_BMP085 bmp;

#include <dht.h>

#define dht_dpin A0 //no ; here. Set equal to channel sensor is on

dht DHT;
float v=0;

int soil=0;
void setup()
{
  Serial.begin(9600);
  delay(200);
  Wire.begin();
  RTC.begin();
  bmp.begin();

  if (! RTC.isrunning()) {
//Serial.println("RTC is NOT running!");
// following line sets the RTC to the date & time this sketch was compiled
RTC.adjust(DateTime(F(__DATE__), F(__TIME__)));}
// This line sets the RTC with an explicit date & time, for example to set
    // January 21, 2014 at 3am you would call:
    // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
}


void loop()
{
   DateTime now = RTC.now();
Serial.print(now.day(), DEC);
Serial.print('/');
Serial.print(now.month(), DEC);
Serial.print('/');
Serial.print(now.year(), DEC);
Serial.print(' ');
delay(200);
//lcd.setCursor(0, 2);
```

```
if (now.hour()<10)
Serial.print('0');
Serial.print(now.hour(), DEC);
Serial.print(':');
if (now.minute()<10)
Serial.print('0');
Serial.print(now.minute(), DEC);
Serial.print(':');
if (now.second()<10)
Serial.print('0');
Serial.print(now.second(), DEC);
delay(200);
Serial.print(" ");
//lcd.setCursor(0, 3);
int dayofweek = now.dayOfWeek();
switch(dayofweek){
case 1:
Serial.print("Monday  ");
break;
case 2:
Serial.print("Tuesday  ");
break;
case 3:
Serial.print("Wednesday  ");
break;
case 4:
Serial.print("Thursday  ");
break;
case 5:
Serial.print("Friday  ");
break;
case 6:
Serial.print("Saturday  ");
break;
case 0:
Serial.print("Sunday  ");
break;
delay(200);
}

Serial.print("Temp. = ");
      Serial.print(bmp.readTemperature());
      Serial.print("C ");
    delay(200);
```

```
      Serial.print("Presr.= ");
      Serial.print(bmp.readPressure());
     Serial.print("Pa ");
   delay(200);
    Serial.print("Altitude = ");
    Serial.print(bmp.readAltitude());
    Serial.print("meters ");
    delay(200);

   DHT.read11(dht_dpin);
Serial.print("Humidity = ");
Serial.print(DHT.humidity);
Serial.print("% ");
delay(200);

v=analogRead(A3);
float a=((v*50*2)/1023);
  float Lux=a*10;
  //float lux= current*2;
   //Serial.println(current);
  Serial.print("Illuminance = ");
  Serial.print(Lux);
  Serial.print("Lux ");
  delay(200);

  int sensorValue = analogRead(A2);
  sensorValue = constrain(sensorValue, 20, 1023);
  soil = map(sensorValue, 20, 1023, 100, 0);

  // print out the soil water percentage you calculated:
  //Serial.print(sensorValue);

  Serial.print("Water content= ");
  Serial.print(soil);
  Serial.println("% ");
  delay(200);
}
```

Wireless and Ethernet Communication

```
#include <Wire.h>
#include "RTClib.h"
```

```
RTC_DS1307 rtc;
DateTime now;

#include <Adafruit_BMP085.h>
Adafruit_BMP085 bmp;

#include <SPI.h>
#include <UIPEthernet.h>

EthernetServer server = EthernetServer(80);

#include "DHT.h"
#define DHTPIN 2     // what pin we're connected to
#define DHTTYPE DHT22   // DHT 22  (AM2302)
DHT dht(DHTPIN, DHTTYPE, 30);

#define lm35_pin A0
#define light_sensor_pin A1
#define moisture_sensor_pin A2

float light = 0;
int sensorValue;
int Temp;
int Pressure;
int Altitude;
int Humidity;
float abcd;
float Lux;

int moisture_content = 0;

void setup()
{
  //Wire.begin();
  Wire1.begin(); // Shield I2C pins connect to alt I2C bus on Arduino Due
  rtc.begin();
 }
  uint8_t mac[6] = {0x00, 0x01, 0x02, 0x03, 0x04, 0x05};
  IPAddress myIP(192, 168, 44, 244);
  Ethernet.begin(mac, myIP);
  server.begin();

  Serial.begin(115200);
  delay(200);
```

```
  bmp.begin();
  dht.begin();
}
void loop()
{
  now = rtc.now();
  SerialClockDisplay();
  get_parameters();
  display_on_tcp();
  display_on_serial();
}
void get_parameters() {
  Temp =  analogRead(lm35_pin);
  delay(1);
  //Temp=(5.0 * analogRead(lm35_pin) * 100.0) / 1024;
  Temp = ((analogRead(lm35_pin) * 3300) / 1024) / 10;
  Pressure = bmp.readPressure();
  Altitude = bmp.readAltitude();
  Humidity = dht.readHumidity();
  //Get Ambient Light Value from Lux Sensor
  light = analogRead(light_sensor_pin);
  delay(1);
  light = analogRead(light_sensor_pin);
  //Get Moisture Content
  int sensorValue = analogRead(moisture_sensor_pin);
  sensorValue = constrain(sensorValue, 20, 1023);
  moisture_content = map(sensorValue, 20, 1023, 100, 0);
}

void display_on_serial() {
  Serial.print("Temp ");
  Serial.print(Temp);
  Serial.print("C ");
  //Get Pressure from BMP085
  Serial.print("Pr= ");
  Serial.print(Pressure);
  Serial.print("Pa ");
  Serial.print("Altitude= ");
  Serial.print(Altitude);
  Serial.print("m ");
  //Get Humidity from DHT11 Sensor
  //DHT.read22(dht_dpin);
  Serial.print("Humidity = ");
  Serial.print(Humidity);
```

```
  Serial.print("% ");
  //float lux= current*2;
  //Serial.println(current);
  Serial.print("Illuminance = ");
  Serial.print(light);
  Serial.print("Lux ");
  // print out the moisture_content water percentage you calculated:
  Serial.print("Moist= ");
  Serial.print(moisture_content);
  Serial.println("% ");
  //digitalWrite(led,LOW);
  //delay(2000);
}
void display_on_tcp() {
  size_t size;

  if (EthernetClient client = server.available())
  {
    while ((size = client.available()) > 0)
    {
      uint8_t* msg = (uint8_t*)malloc(size);
      size = client.read(msg, size);
      Serial.write(msg, size);
      free(msg);
    }
    // send a standard http response header
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("Connection: close");  // the connection will be closed after complet
    client.println("Refresh: 5");  // refresh the page automatically every 5 sec
    client.println();
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    // output the value of each analog input pin
    // digital clock display of the time

    client.print(now.year(), DEC);
    client.print('/');
    client.print(now.month(), DEC);
    client.print('/');
    client.print(now.day(), DEC);
    client.print(' ');
    client.print(now.hour(), DEC);
    client.print(':');
```

```
    client.print(now.minute(), DEC);
    client.print(':');
    client.print(now.second(), DEC);
    client.print("  ");
    //remove the following line to print a string on TCP
    client.println("<br />");
    //Get Temperature from LM35
    client.print("Temp ");
    client.print(Temp);
    client.print("*C ");
    //remove the following line to print a string on TCP
    client.println("<br />");
    //Get Pressure from BMP085
    client.print("Pr= ");
    client.print(Pressure);
    client.print("Pa ");
    client.print("Altitude= ");
    client.print(Altitude);
    client.print("m ");
    //remove the following line to print a string on TCP
    client.println("<br />");
    //Get Humidity from DHT11 Sensor
    //DHT.read22(dht_dpin);
    client.print("Humidity = ");
    client.print(Humidity);
    client.print("% ");
    //remove the following line to print a string on TCP
    client.println("<br />");
    //float lux= current*2;
    //client.println(current);
    client.print("Illuminance = ");
    client.print(light);
    client.print("Lux ");
    //remove the following line to print a string on TCP
    client.println("<br />");
    // print out the moisture_content water percentage you calculated:
    client.print("Moist= ");
    client.print(moisture_content);
    client.println("% ");
    client.println("</html>");
    client.stop();
  }
}
```

```
void SerialClockDisplay() {
  // digital clock display of the time
  now = rtc.now();
  Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.day(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.print("  ");
}
void printDigits(int digits) {
  Serial.print(":");
  if (digits < 10)
    Serial.print('0');
  Serial.print(digits);
}
```